FIG. 1

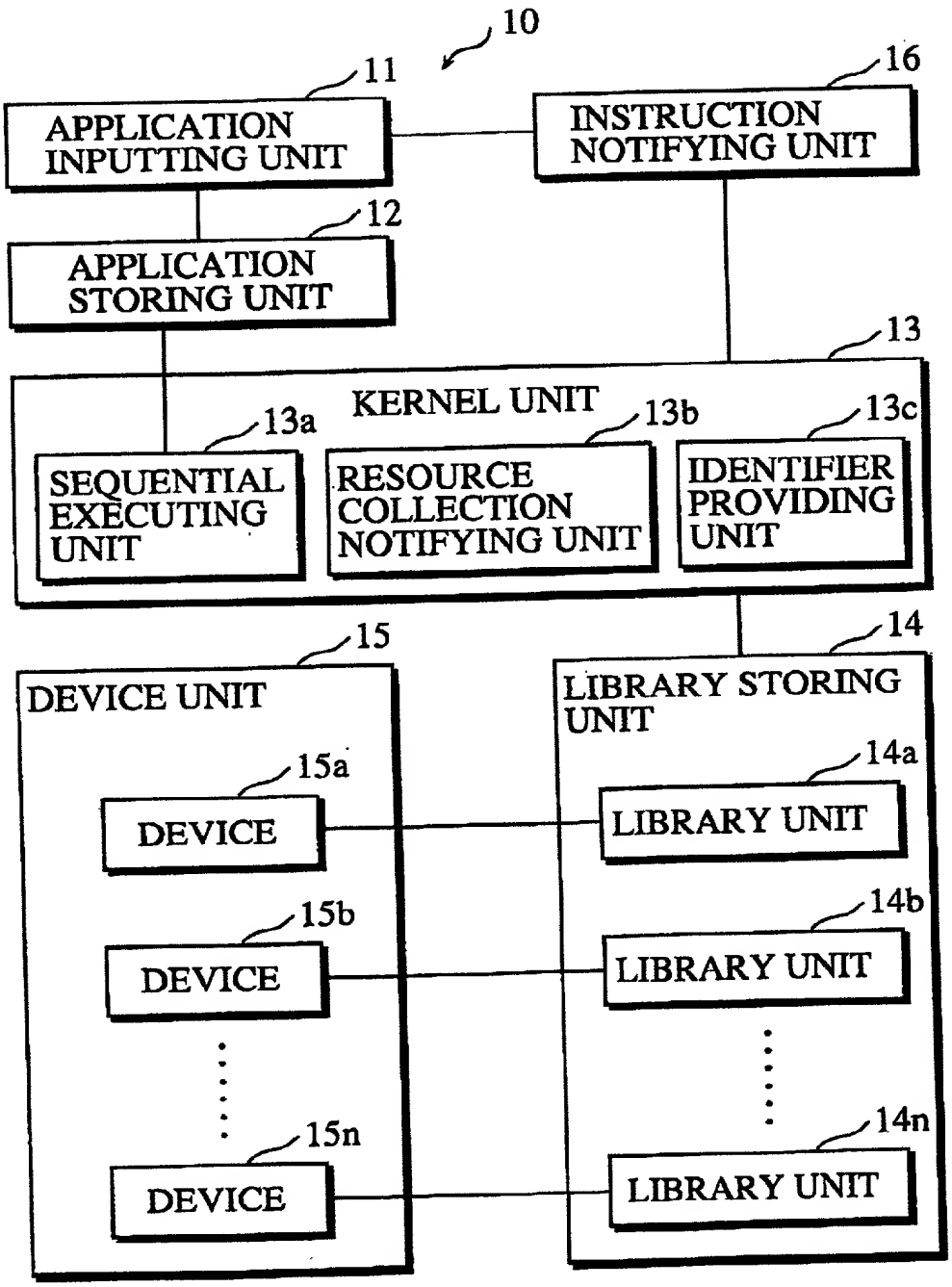

10
11
APPLICATION INPUTTING UNIT
16
INSTRUCTION NOTIFYING UNIT
12
APPLICATION STORING UNIT
13
KERNEL UNIT
13a
SEQUENTIAL EXECUTING UNIT
13b
RESOURCE COLLECTION NOTIFYING UNIT
13c
IDENTIFIER PROVIDING UNIT
15
DEVICE UNIT
14
LIBRARY STORING UNIT
15a
DEVICE
14a
LIBRARY UNIT
15b
DEVICE
14b
LIBRARY UNIT
15n
DEVICE
14n
LIBRARY UNIT

FIG. 2

```
typedef void(*CALLBACK_f)(int Appid)
```

FIG. 3

```
1     CALLBACK_f callf[100];
2
3     void init() {
4        for (I-0;I<100; I++ ) callf[I]=null;
5     }
6     int add_callf(CALLBACK_f callfunc){
7        for (I=0; I<100; I++ ){
8           if ( callf[I]==null) {
9              callf[I] = callfunc;
10             return(1);
11          }
12       }
13       return(-1);
14    }
15
16    void remove_callf(CALLBACK_f callfunc){
17       for (I=0; I<100; I++){
18          if ( callf[I]==callfunc ) callf[I] = null;
19       }
20    }
21
22    void call_callback (int Appid) {
23       for(I=0; I<100; I++ ){
24          if ( (callf[I] !=null) (*callf[I])(Appid);
25       }
26    }
```

FIG. 4

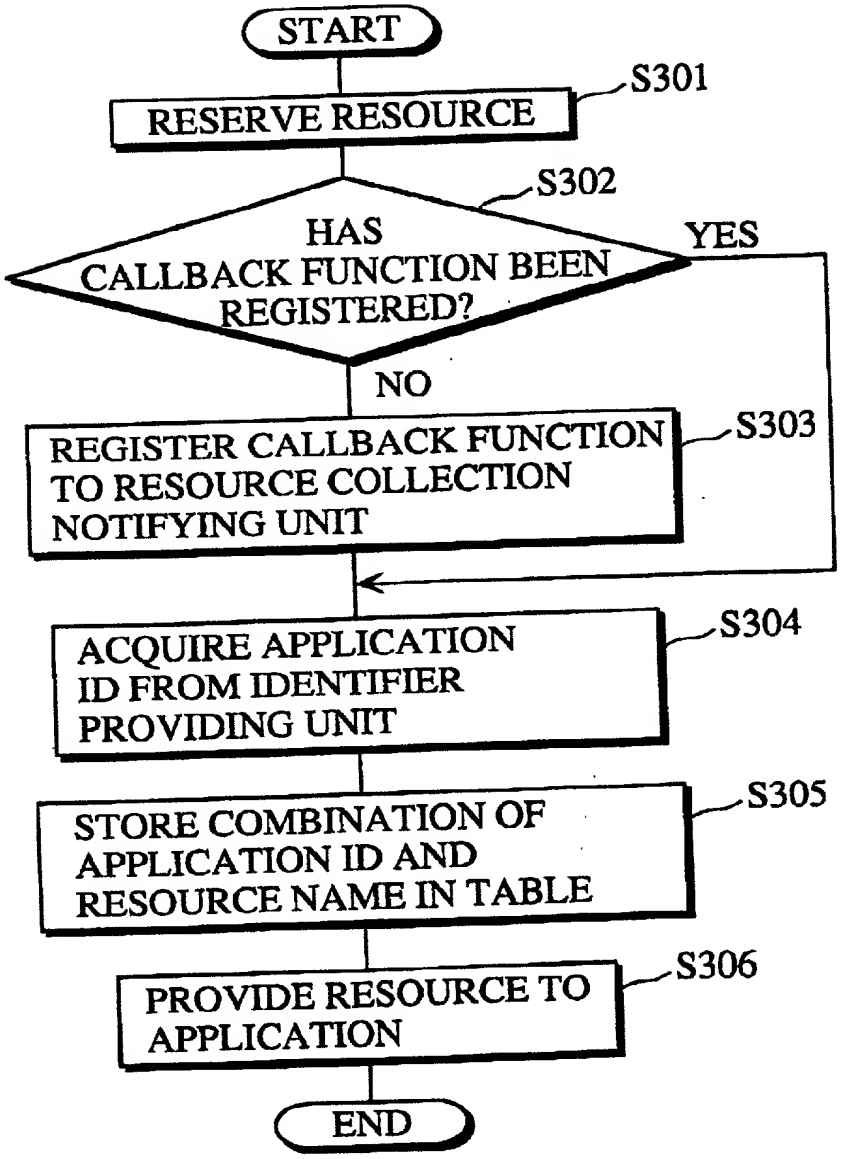

START
RESERVE RESOURCE
S301
S302
HAS CALLBACK FUNCTION BEEN REGISTERED?
YES
NO
REGISTER CALLBACK FUNCTION TO RESOURCE COLLECTION NOTIFYING UNIT
S303
ACQUIRE APPLICATION ID FROM IDENTIFIER PROVIDING UNIT
S304
STORE COMBINATION OF APPLICATION ID AND RESOURCE NAME IN TABLE
S305
PROVIDE RESOURCE TO APPLICATION
S306
END

FIG. 5

(1)

APPLICATION ID	FILENAME
1	a.txt

(2)

APPLICATION ID	FILENAME
1	a.txt
2	b.txt

(3)

APPLICATION ID	FILENAME
1	a.txt

FIG. 6

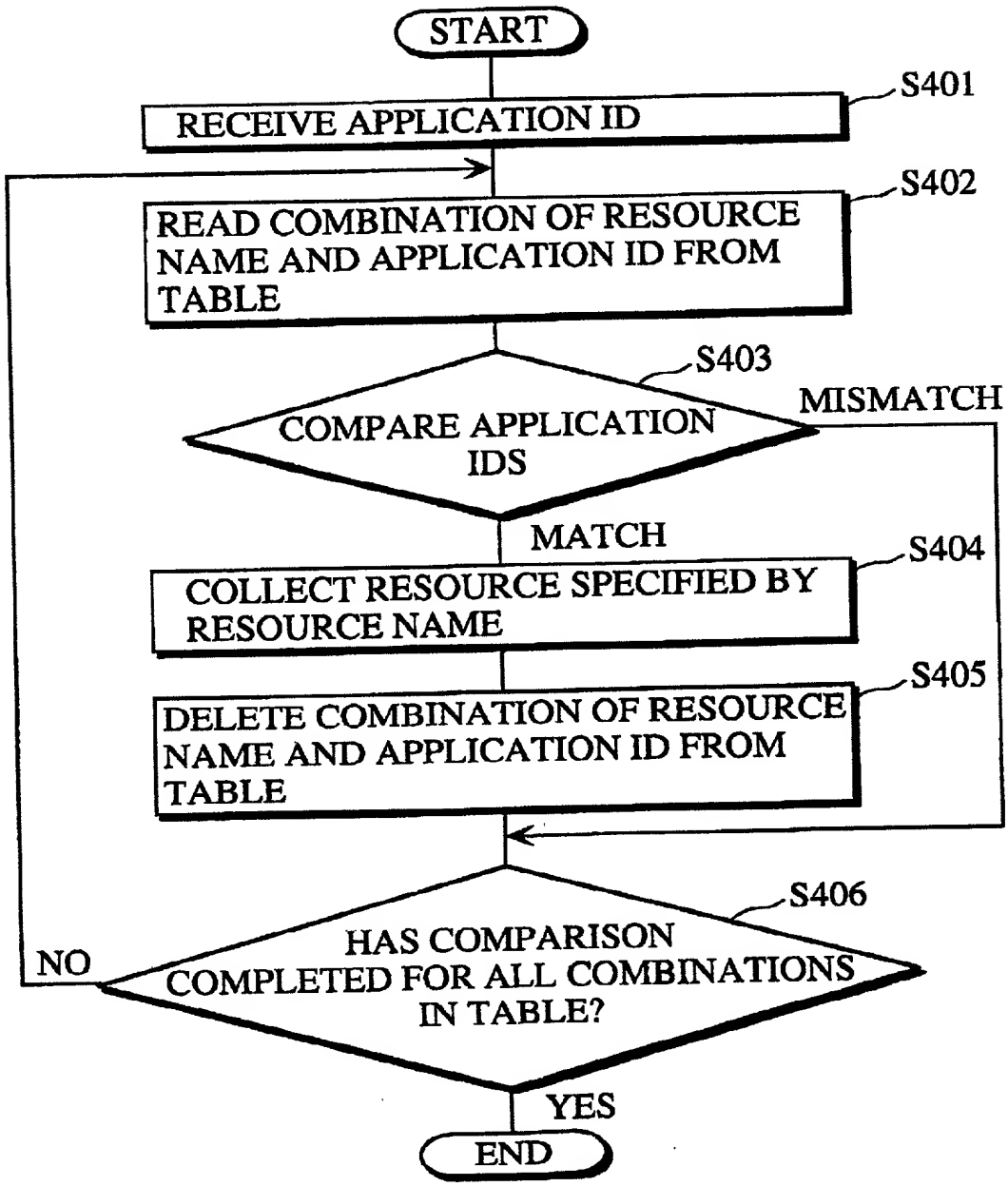

START
RECEIVE APPLICATION ID
S401
READ COMBINATION OF RESOURCE NAME AND APPLICATION ID FROM TABLE
S402
COMPARE APPLICATION IDS
S403
MISMATCH
MATCH
COLLECT RESOURCE SPECIFIED BY RESOURCE NAME
S404
DELETE COMBINATION OF RESOURCE NAME AND APPLICATION ID FROM TABLE
S405
HAS COMPARISON COMPLETED FOR ALL COMBINATIONS IN TABLE?
S406
NO
YES
END

FIG. 7

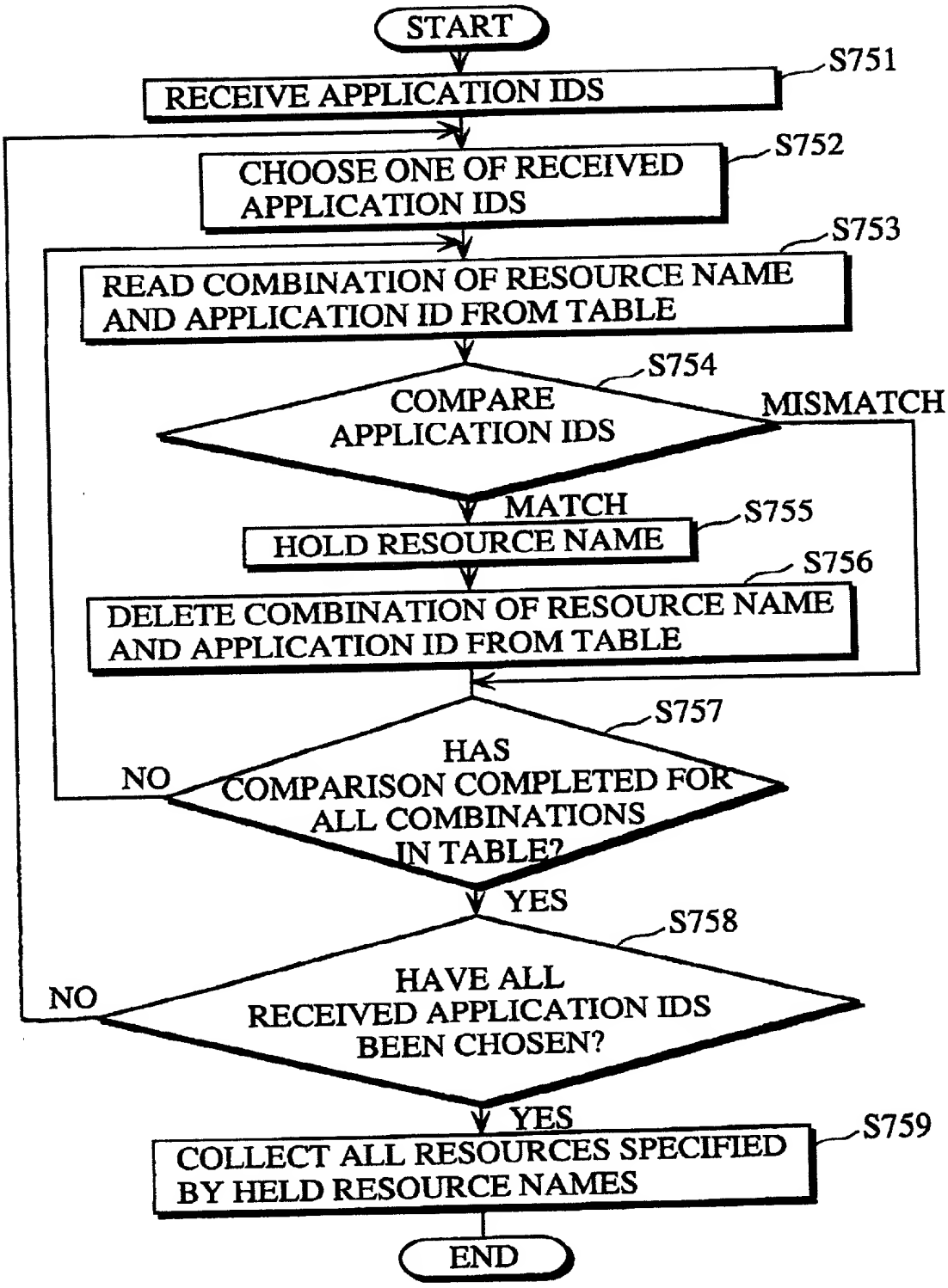

START
RECEIVE APPLICATION IDS
S751
CHOOSE ONE OF RECEIVED APPLICATION IDS
S752
READ COMBINATION OF RESOURCE NAME AND APPLICATION ID FROM TABLE
S753
COMPARE APPLICATION IDS
S754
MISMATCH
MATCH
HOLD RESOURCE NAME
S755
DELETE COMBINATION OF RESOURCE NAME AND APPLICATION ID FROM TABLE
S756
HAS COMPARISON COMPLETED FOR ALL COMBINATIONS IN TABLE?
S757
NO
YES
HAVE ALL RECEIVED APPLICATION IDS BEEN CHOSEN?
S758
NO
YES
COLLECT ALL RESOURCES SPECIFIED BY HELD RESOURCE NAMES
S759
END

FIG. 8

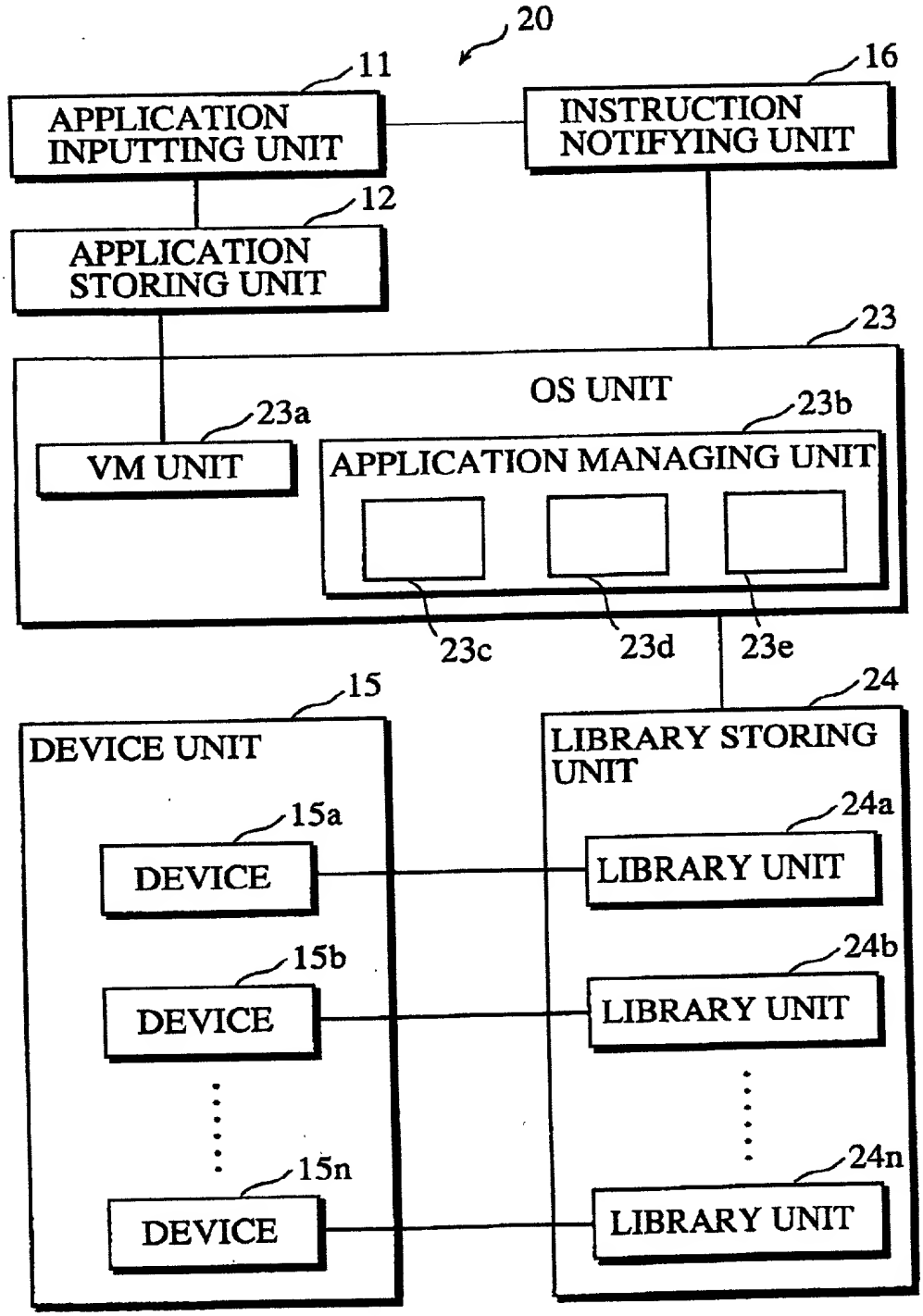

20
11
APPLICATION INPUTTING UNIT
16
INSTRUCTION NOTIFYING UNIT
12
APPLICATION STORING UNIT
23
OS UNIT
23a
VM UNIT
23b
APPLICATION MANAGING UNIT
23c
23d
23e
15
DEVICE UNIT
24
LIBRARY STORING UNIT
15a
DEVICE
24a
LIBRARY UNIT
15b
DEVICE
24b
LIBRARY UNIT
15n
DEVICE
24n
LIBRARY UNIT

FIG. 9

```
public interface resourceCollectionListener{
   public void update();
}
```

FIG. 10

```
public class applicationProxy {
   public string name;  // application name

   public int getStatus() {
      // get the status of application,paues or destroy
   }

   public addListener(resourceCollectionListener 1){
      // register resource collection instance
   }

   public removeListener(resourceCollectionListener 1){
     // remove resource collection intance
   }
}
```

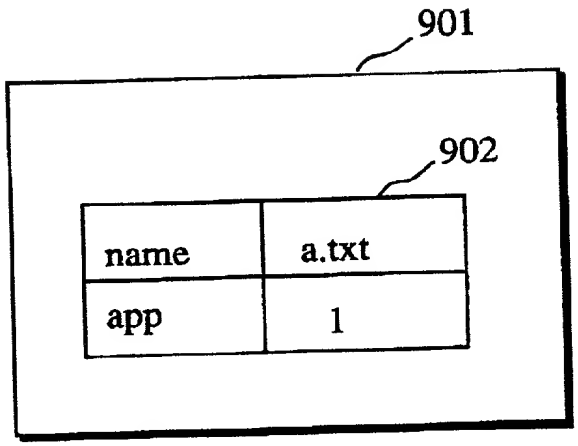
FIG. 11

```
public class fileCollectionListener
            implements resourceCollectionListener{
   String name;  // filename
   ApplicationProxy app;
                        // Application information instance

   public void update() {
      // release file which has filename
   }
}
```

FIG. 12

(1)


901
902
name
a.txt
app
1

(2)

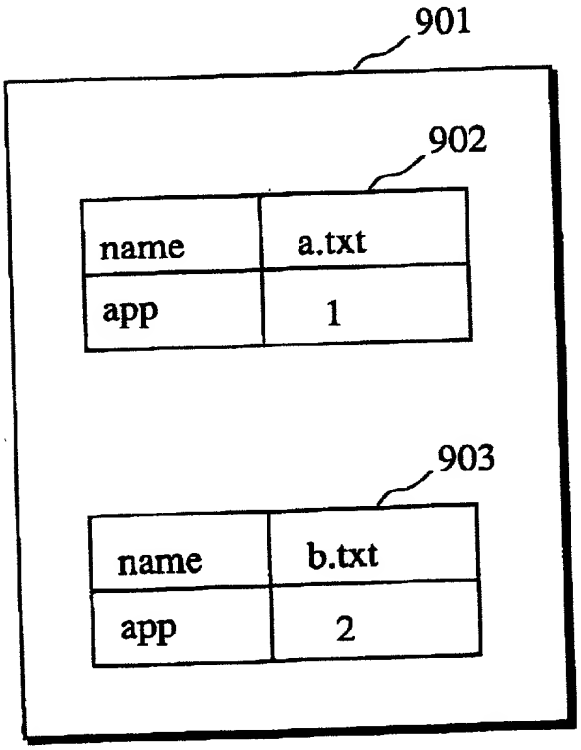

901
902
name
a.txt
app
1
903
name
b.txt
app
2

FIG. 13

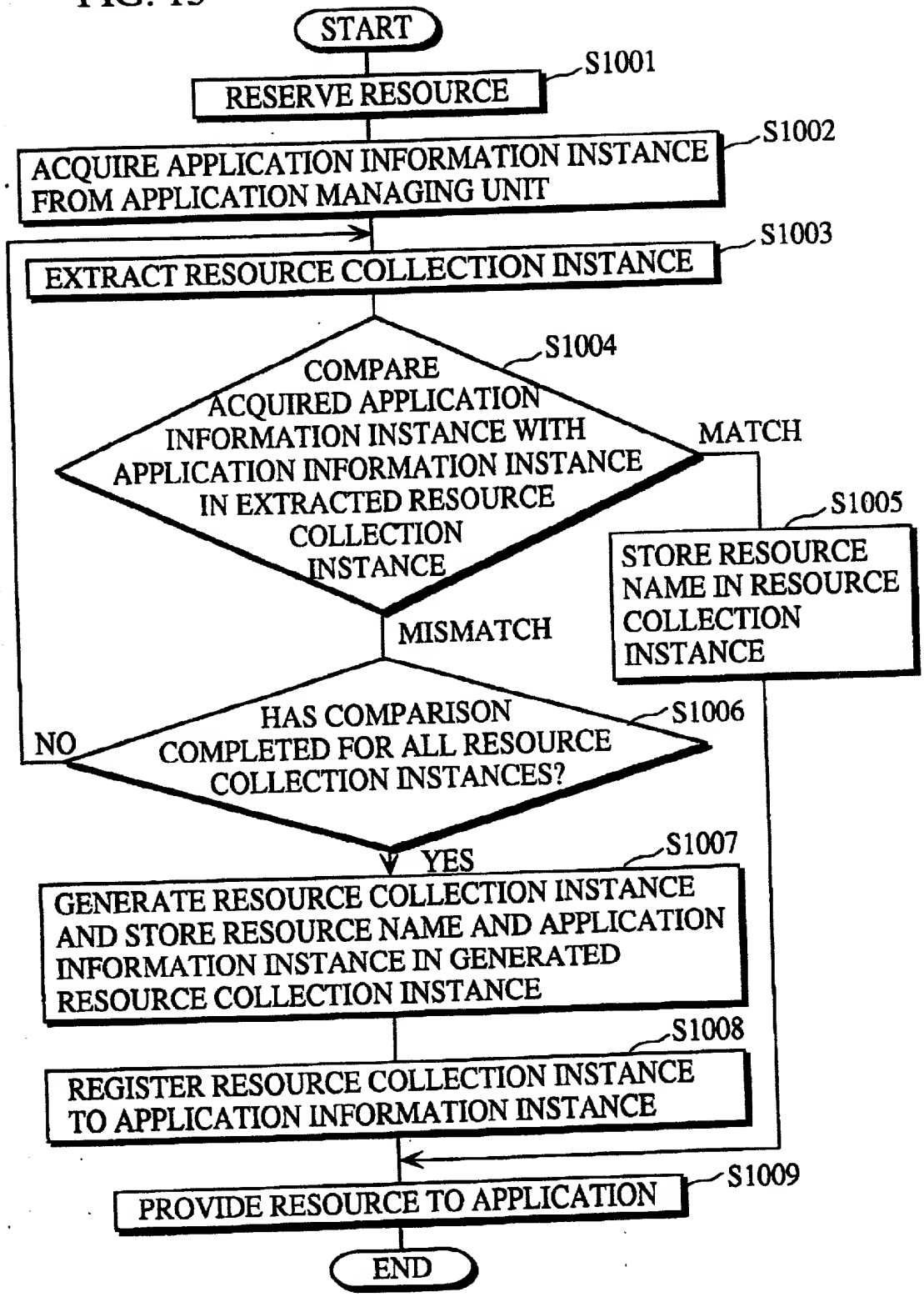

START
RESERVE RESOURCE
S1001
ACQUIRE APPLICATION INFORMATION INSTANCE FROM APPLICATION MANAGING UNIT
S1002
EXTRACT RESOURCE COLLECTION INSTANCE
S1003
COMPARE ACQUIRED APPLICATION INFORMATION INSTANCE WITH APPLICATION INFORMATION INSTANCE IN EXTRACTED RESOURCE COLLECTION INSTANCE
S1004
MATCH
MISMATCH
STORE RESOURCE NAME IN RESOURCE COLLECTION INSTANCE
S1005
HAS COMPARISON COMPLETED FOR ALL RESOURCE COLLECTION INSTANCES?
S1006
NO
YES
GENERATE RESOURCE COLLECTION INSTANCE AND STORE RESOURCE NAME AND APPLICATION INFORMATION INSTANCE IN GENERATED RESOURCE COLLECTION INSTANCE
S1007
REGISTER RESOURCE COLLECTION INSTANCE TO APPLICATION INFORMATION INSTANCE
S1008
PROVIDE RESOURCE TO APPLICATION
S1009
END

FIG. 14

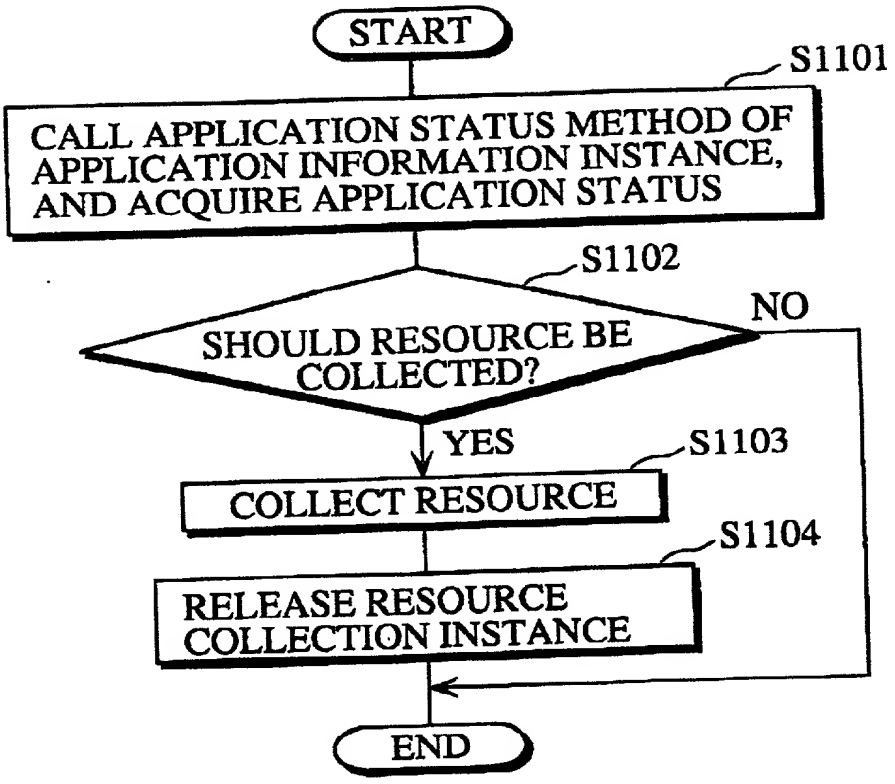

START
S1101
CALL APPLICATION STATUS METHOD OF APPLICATION INFORMATION INSTANCE, AND ACQUIRE APPLICATION STATUS
S1102
SHOULD RESOURCE BE COLLECTED?
NO
YES
S1103
COLLECT RESOURCE
S1104
RELEASE RESOURCE COLLECTION INSTANCE
END

FIG. 15

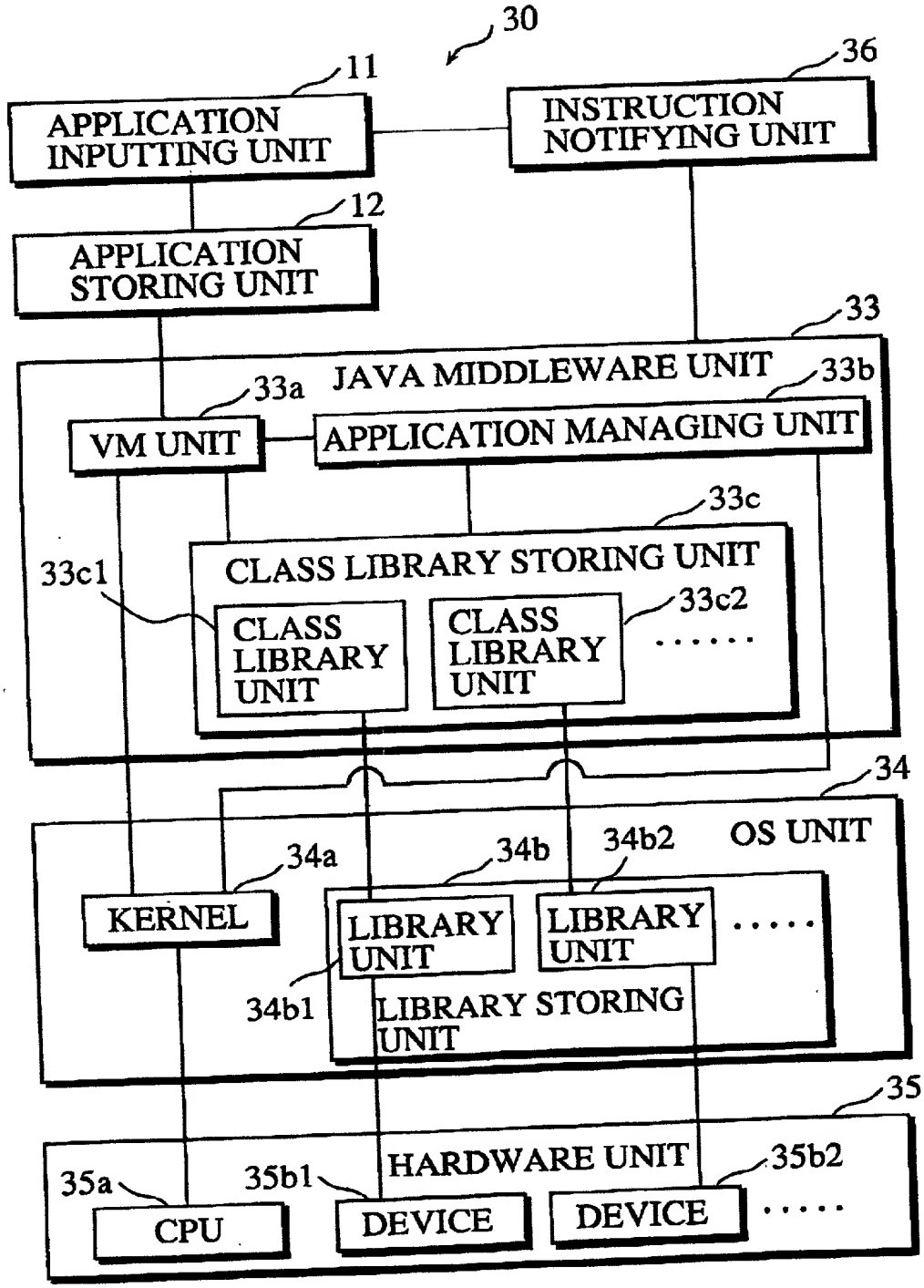

30
11
APPLICATION INPUTTING UNIT
36
INSTRUCTION NOTIFYING UNIT
12
APPLICATION STORING UNIT
33
JAVA MIDDLEWARE UNIT
33a
VM UNIT
33b
APPLICATION MANAGING UNIT
33c
33c1
CLASS LIBRARY STORING UNIT
CLASS LIBRARY UNIT
33c2
CLASS LIBRARY UNIT
.....
34
OS UNIT
34a
KERNEL
34b
34b2
LIBRARY UNIT
LIBRARY UNIT
.....
34b1
LIBRARY STORING UNIT
35
HARDWARE UNIT
35a
CPU
35b1
DEVICE
35b2
DEVICE
.....

FIG. 16

(1)

APPLICATION ID	TASK ID	THREAD ID
1	201	1,2
2	202	4,5,6

(2)

APPLICATION ID	TASK ID	THREAD ID
1	201	1,2,7
2	202	4,5,6

(3)

APPLICATION ID	TASK ID	THREAD ID
1	201	1,2,7
2	202	4,6

FIG. 17

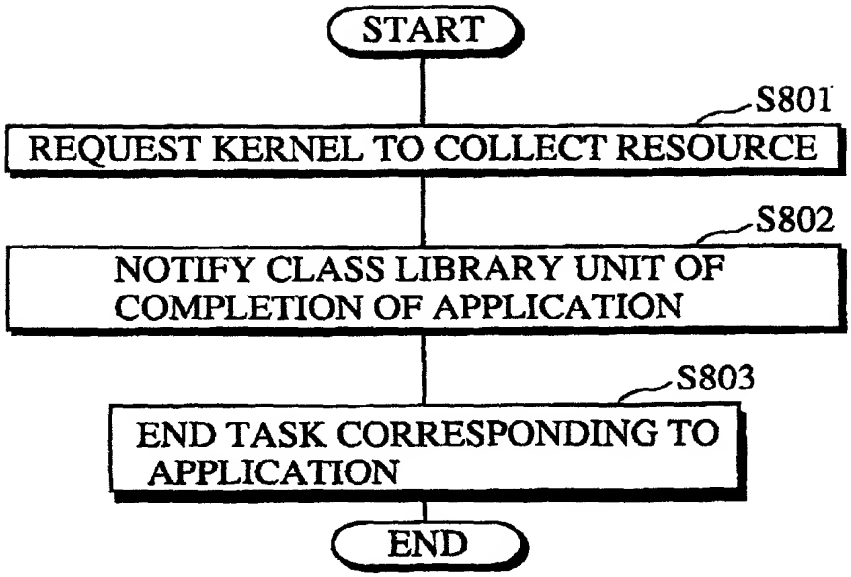

START
S801
REQUEST KERNEL TO COLLECT RESOURCE
S802
NOTIFY CLASS LIBRARY UNIT OF COMPLETION OF APPLICATION
S803
END TASK CORRESPONDING TO APPLICATION
END

FIG. 18

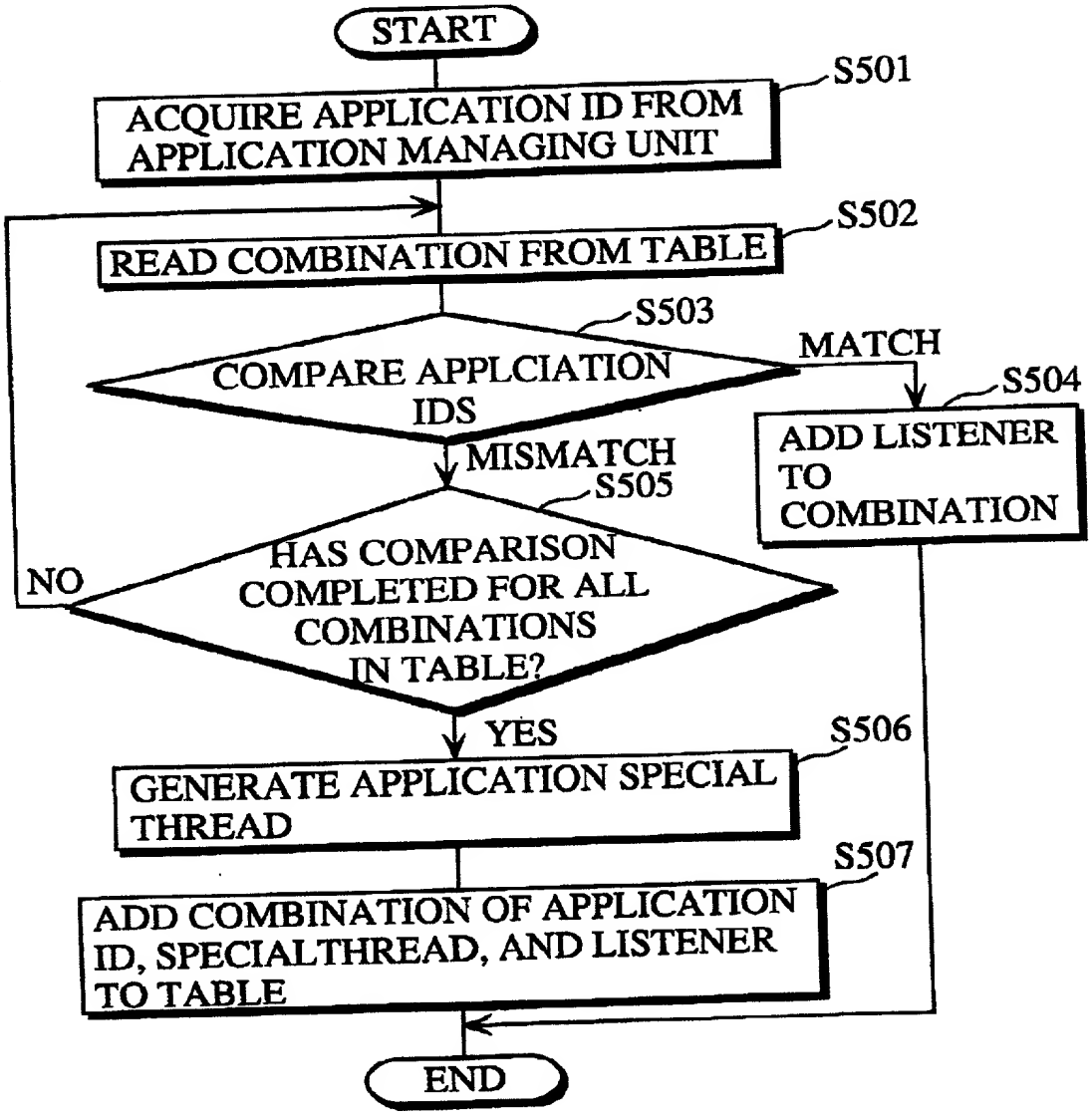

START
ACQUIRE APPLICATION ID FROM APPLICATION MANAGING UNIT
S501
READ COMBINATION FROM TABLE
S502
COMPARE APPLCIATION IDS
S503
MATCH
ADD LISTENER TO COMBINATION
S504
MISMATCH
HAS COMPARISON COMPLETED FOR ALL COMBINATIONS IN TABLE?
S505
NO
YES
GENERATE APPLICATION SPECIAL THREAD
S506
ADD COMBINATION OF APPLICATION ID, SPECIALTHREAD, AND LISTENER TO TABLE
S507
END

FIG. 19

(1)

APPLICATION ID	THREAD INSTANCE	LISTENER INSTANCE
1	thread10	L1,L4

(2)

APPLICATION ID	THREAD INSTANCE	LISTENER INSTANCE
1	thread10	L1,L4
2	thread20	L6

(3)

APPLICATION ID	THREAD INSTANCE	LISTENER INSTANCE
2	thread20	L6

FIG. 20

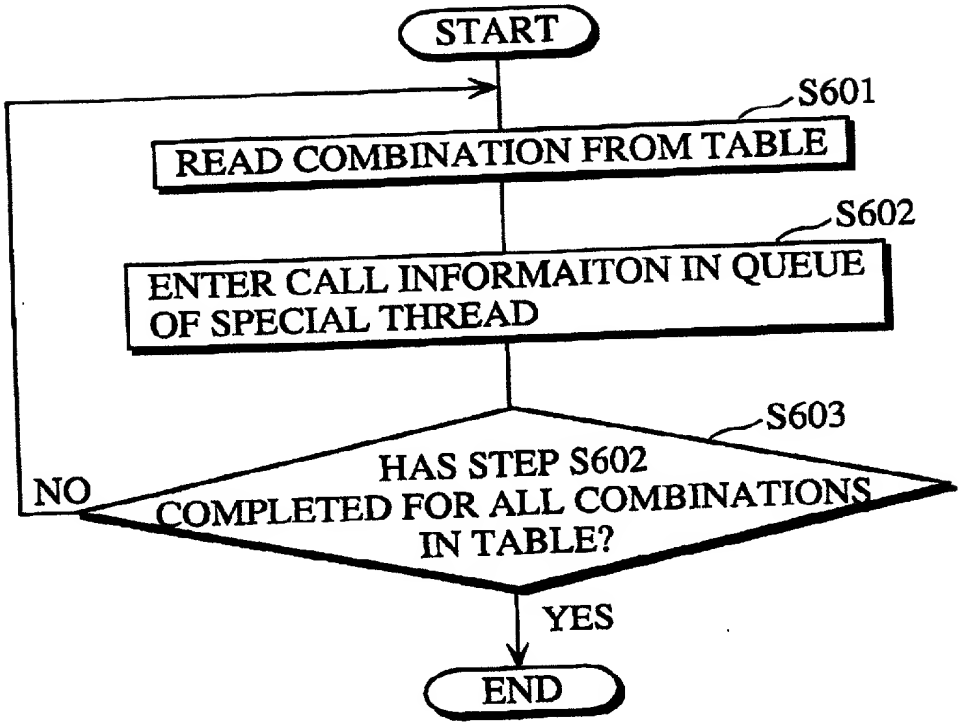

START
S601
READ COMBINATION FROM TABLE
S602
ENTER CALL INFORMAITON IN QUEUE OF SPECIAL THREAD
S603
HAS STEP S602 COMPLETED FOR ALL COMBINATIONS IN TABLE?
NO
YES
END

FIG. 21

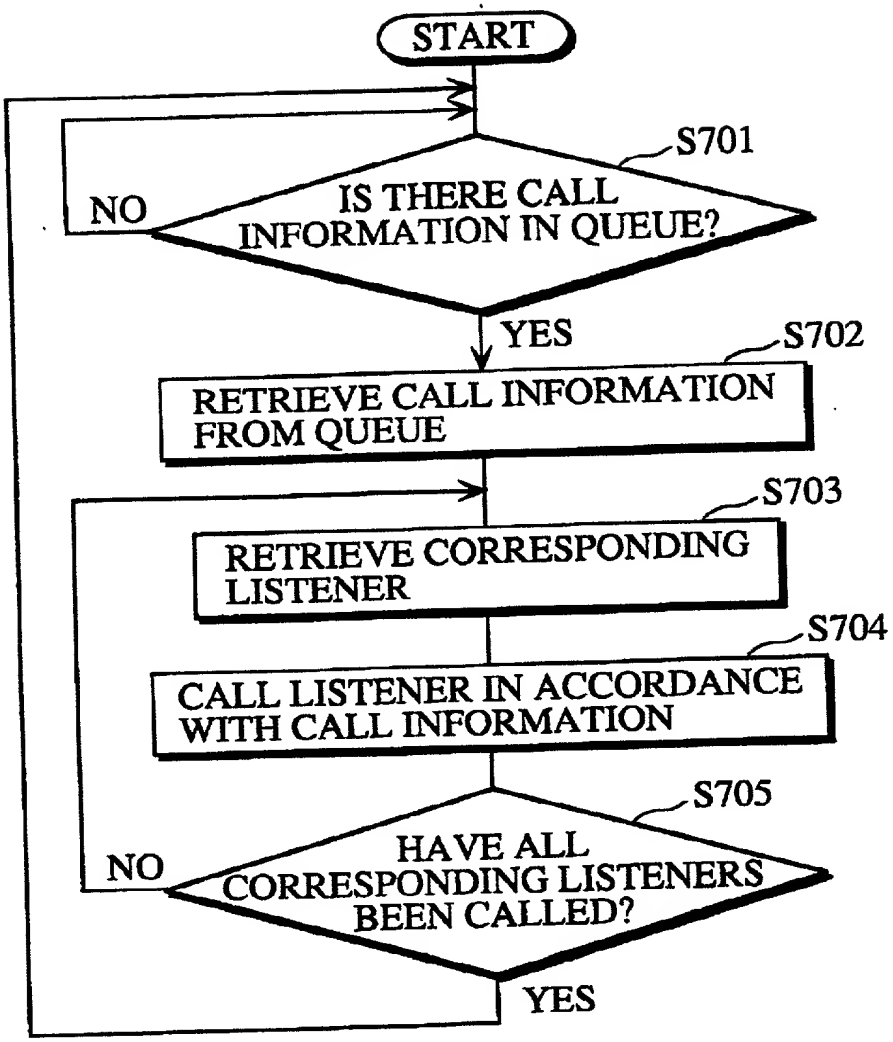

START
S701
IS THERE CALL INFORMATION IN QUEUE?
NO
YES
S702
RETRIEVE CALL INFORMATION FROM QUEUE
S703
RETRIEVE CORRESPONDING LISTENER
S704
CALL LISTENER IN ACCORDANCE WITH CALL INFORMATION
S705
HAVE ALL CORRESPONDING LISTENERS BEEN CALLED?
NO
YES

FIG. 22

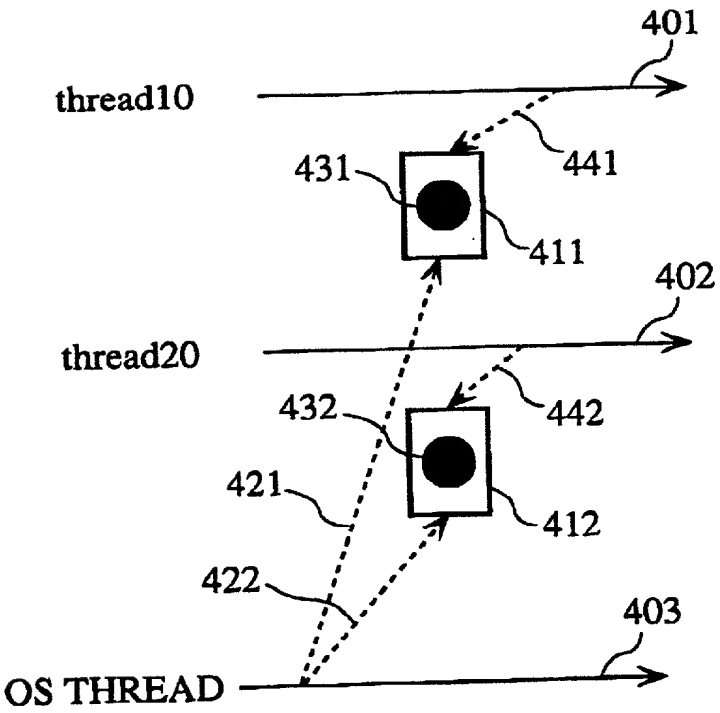

401
thread10
441
431
411
402
thread20
442
432
421
412
422
403
OS THREAD

FIG. 23

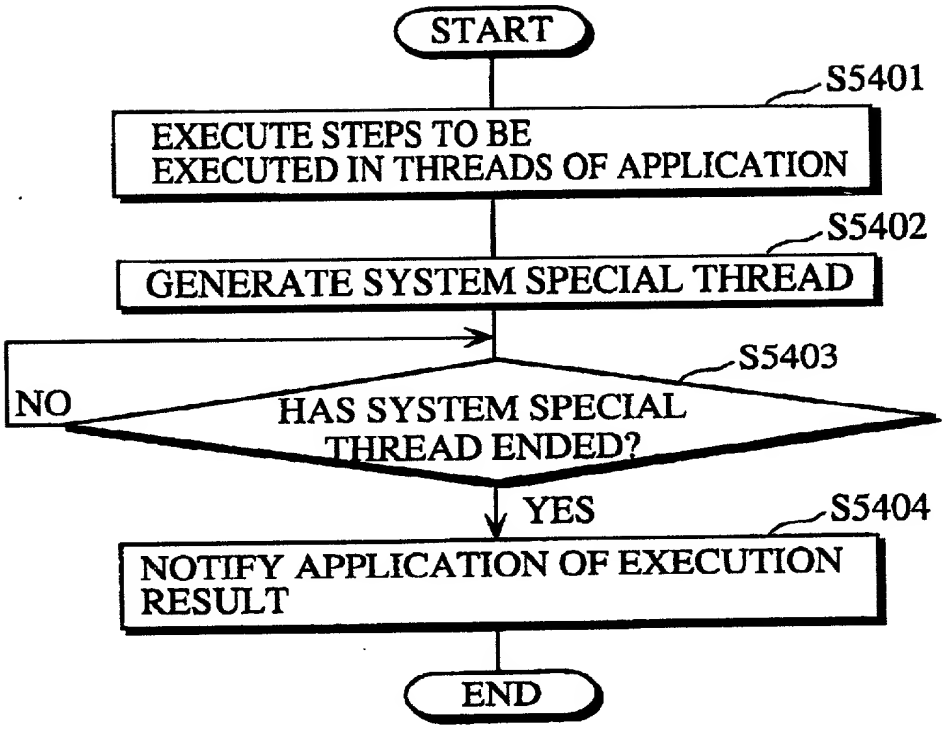

START
S5401
EXECUTE STEPS TO BE EXECUTED IN THREADS OF APPLICATION
S5402
GENERATE SYSTEM SPECIAL THREAD
S5403
NO
HAS SYSTEM SPECIAL THREAD ENDED?
YES
S5404
NOTIFY APPLICATION OF EXECUTION RESULT
END

FIG. 24

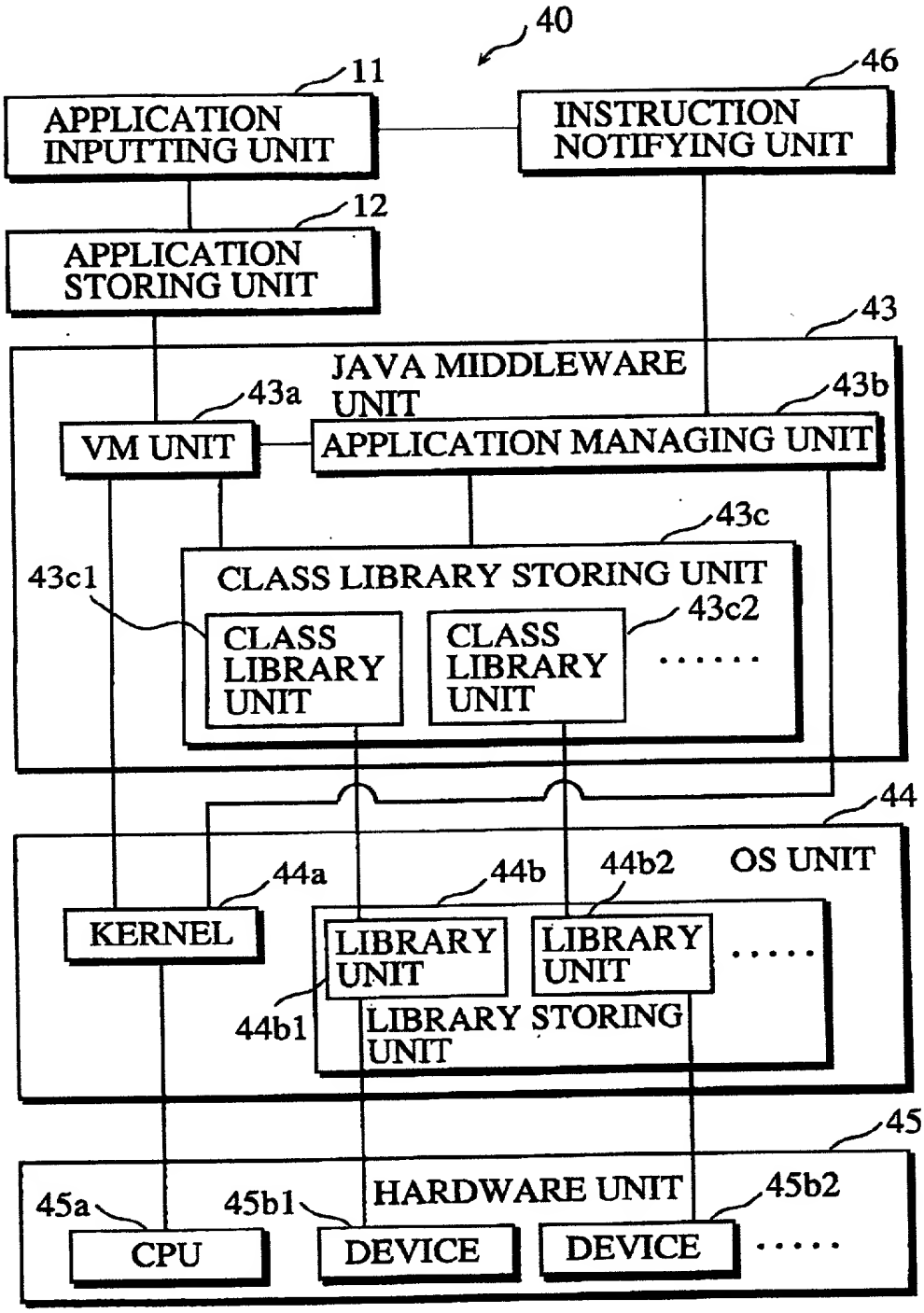

40
11
APPLICATION INPUTTING UNIT
46
INSTRUCTION NOTIFYING UNIT
12
APPLICATION STORING UNIT
43
JAVA MIDDLEWARE UNIT
43a
VM UNIT
43b
APPLICATION MANAGING UNIT
43c
43c1
CLASS LIBRARY STORING UNIT
CLASS LIBRARY UNIT
43c2
CLASS LIBRARY UNIT
.....
44
OS UNIT
44a
KERNEL
44b
LIBRARY UNIT
44b2
LIBRARY UNIT
.....
44b1
LIBRARY STORING UNIT
45
HARDWARE UNIT
45a
CPU
45b1
DEVICE
45b2
DEVICE
.....

FIG. 25

APPLICATION ID	CLASS LOADER
1	classloader1
2	classloader2

FIG. 26

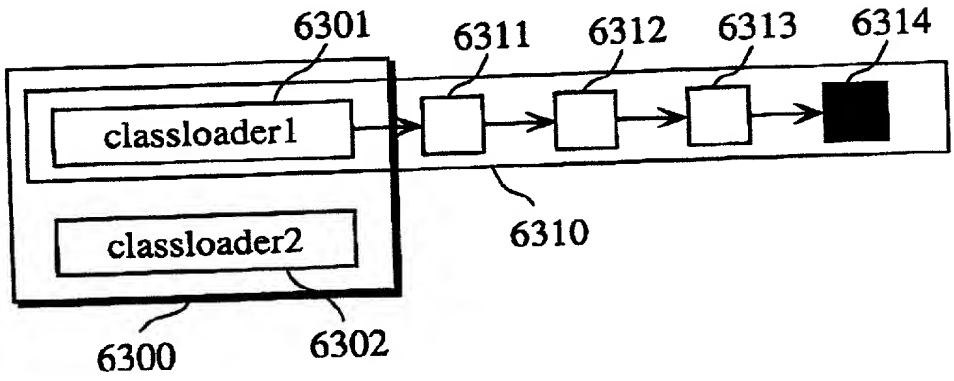


FIG. 27

APPLICATION ID	FILE
1	A.txt B.txt
2	C.txt

FIG. 28

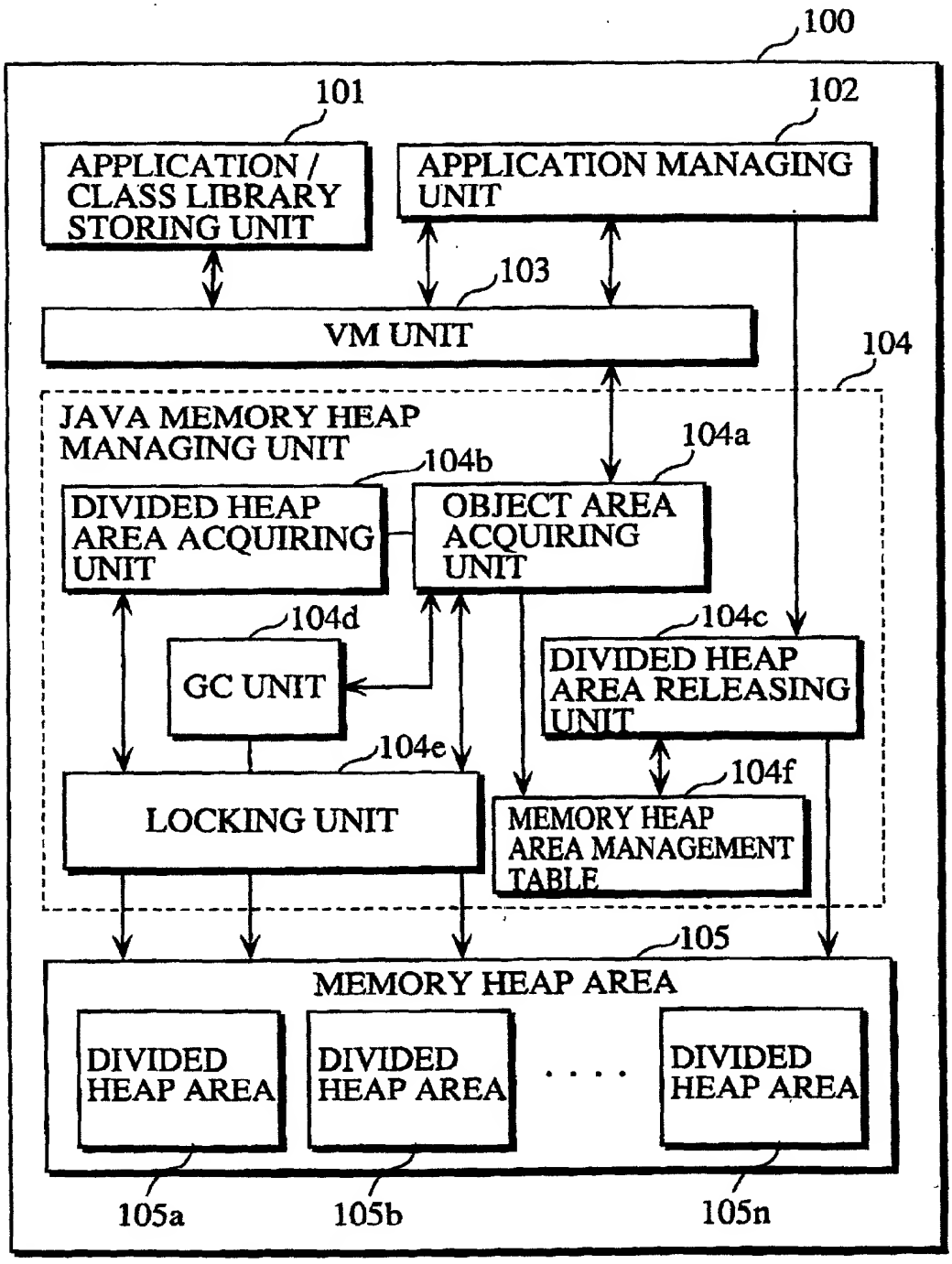

100
101
APPLICATION / CLASS LIBRARY STORING UNIT
102
APPLICATION MANAGING UNIT
103
VM UNIT
104
JAVA MEMORY HEAP MANAGING UNIT
104a
104b
DIVIDED HEAP AREA ACQUIRING UNIT
OBJECT AREA ACQUIRING UNIT
104d
GC UNIT
104c
DIVIDED HEAP AREA RELEASING UNIT
104e
LOCKING UNIT
104f
MEMORY HEAP AREA MANAGEMENT TABLE
105
MEMORY HEAP AREA
DIVIDED HEAP AREA
DIVIDED HEAP AREA
. . . .
DIVIDED HEAP AREA
105a
105b
105n

FIG. 29

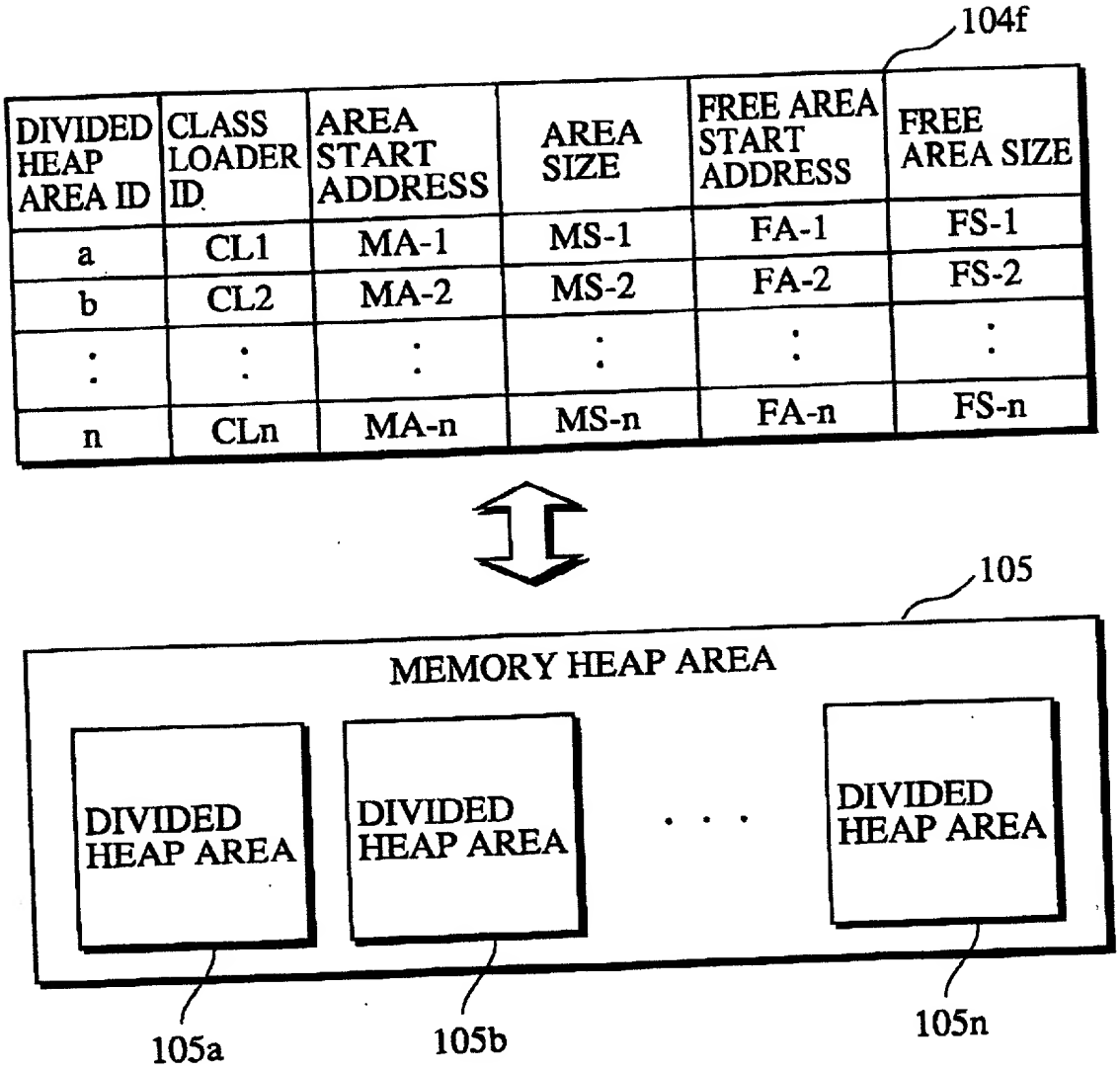

104f
DIVIDED HEAP AREA ID
CLASS LOADER ID
AREA START ADDRESS
AREA SIZE
FREE AREA START ADDRESS
FREE AREA SIZE
a
CL1
MA-1
MS-1
FA-1
FS-1
b
CL2
MA-2
MS-2
FA-2
FS-2
n
CLn
MA-n
MS-n
FA-n
FS-n
105
MEMORY HEAP AREA
DIVIDED HEAP AREA
DIVIDED HEAP AREA
DIVIDED HEAP AREA
105a
105b
105n

FIG. 30

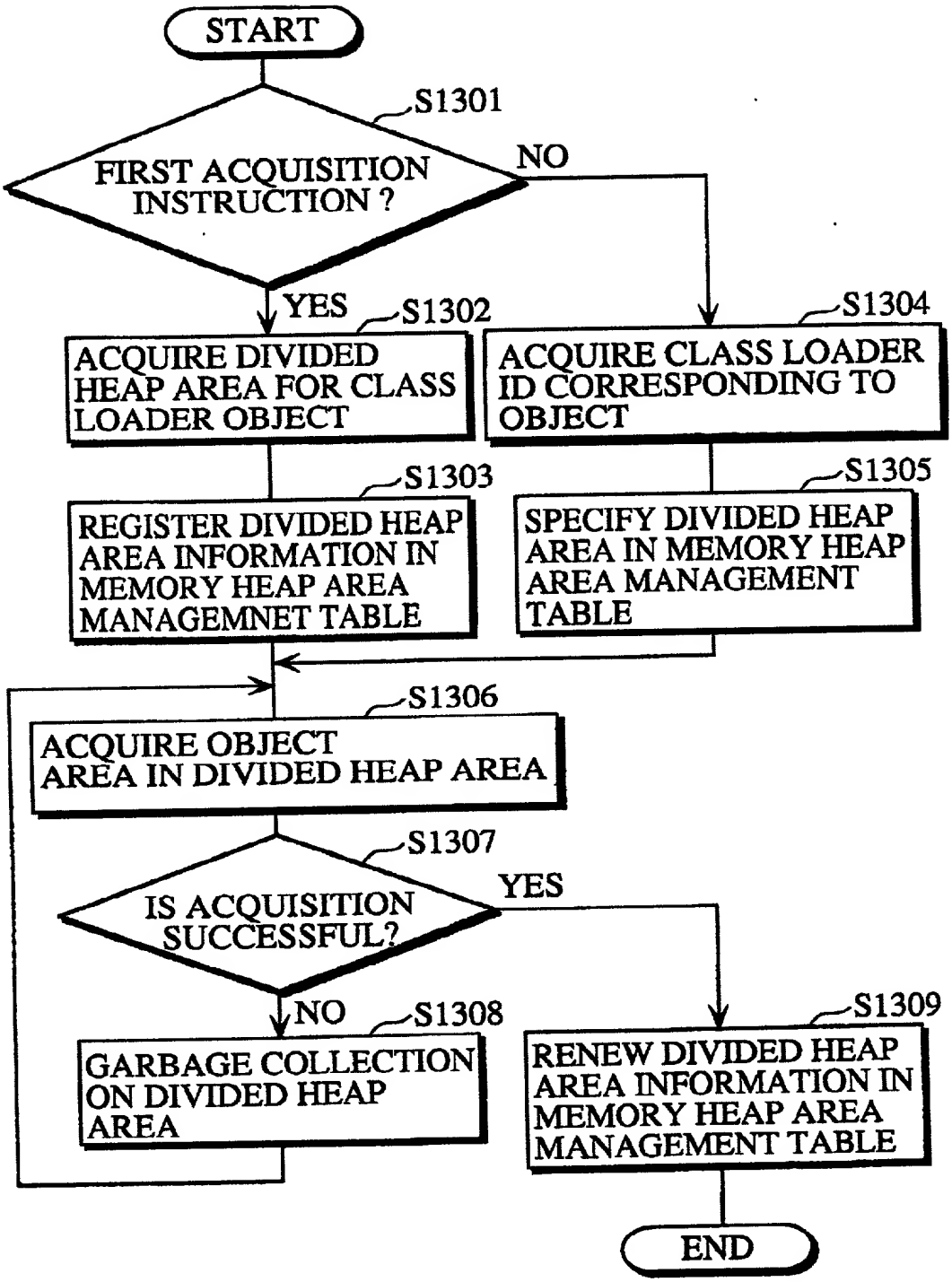

START
S1301
FIRST ACQUISITION INSTRUCTION ?
NO
YES
S1302
ACQUIRE DIVIDED HEAP AREA FOR CLASS LOADER OBJECT
S1304
ACQUIRE CLASS LOADER ID CORRESPONDING TO OBJECT
S1303
REGISTER DIVIDED HEAP AREA INFORMATION IN MEMORY HEAP AREA MANAGEMNET TABLE
S1305
SPECIFY DIVIDED HEAP AREA IN MEMORY HEAP AREA MANAGEMENT TABLE
S1306
ACQUIRE OBJECT AREA IN DIVIDED HEAP AREA
S1307
IS ACQUISITION SUCCESSFUL?
YES
NO
S1308
GARBAGE COLLECTION ON DIVIDED HEAP AREA
S1309
RENEW DIVIDED HEAP AREA INFORMATION IN MEMORY HEAP AREA MANAGEMENT TABLE
END

FIG. 31

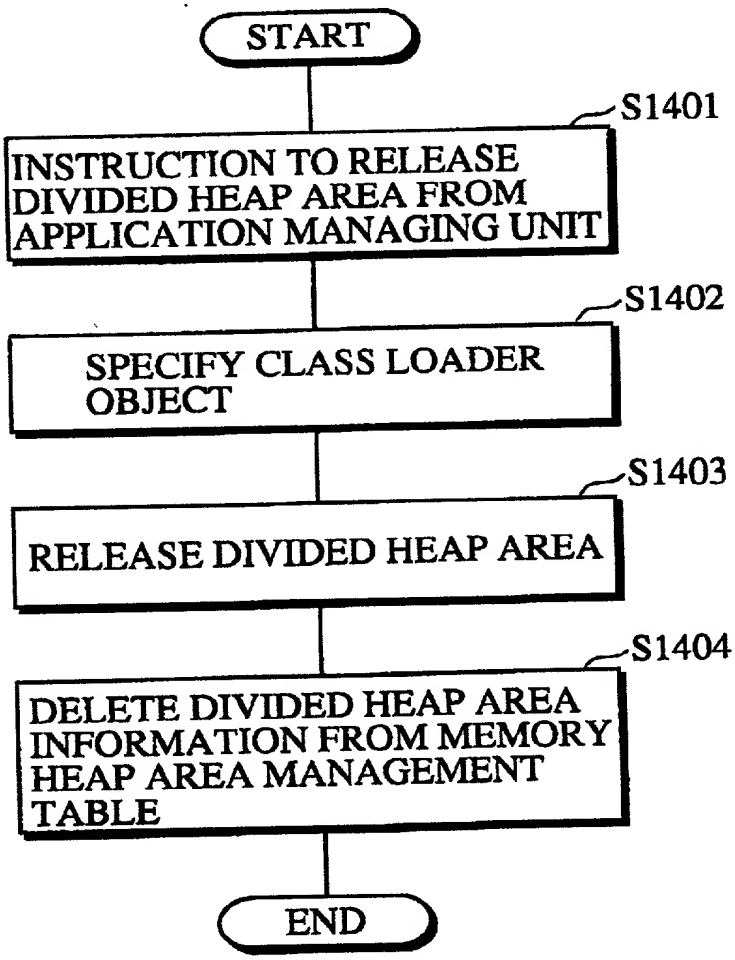
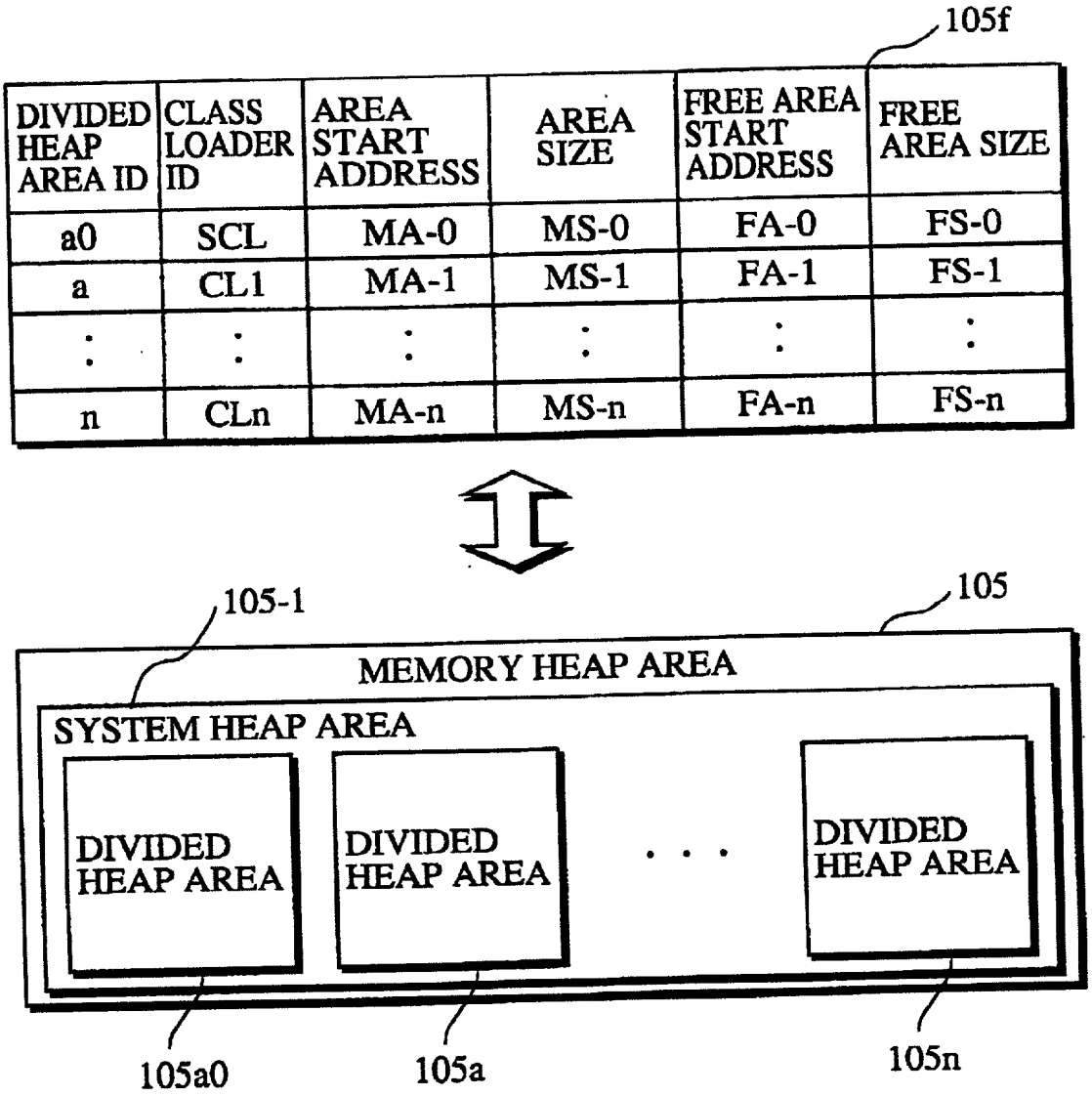

START
S1401
INSTRUCTION TO RELEASE DIVIDED HEAP AREA FROM APPLICATION MANAGING UNIT
S1402
SPECIFY CLASS LOADER OBJECT
S1403
RELEASE DIVIDED HEAP AREA
S1404
DELETE DIVIDED HEAP AREA INFORMATION FROM MEMORY HEAP AREA MANAGEMENT TABLE
END

FIG. 32


105f
DIVIDED HEAP AREA ID
CLASS LOADER ID
AREA START ADDRESS
AREA SIZE
FREE AREA START ADDRESS
FREE AREA SIZE
a0
SCL
MA-0
MS-0
FA-0
FS-0
a
CL1
MA-1
MS-1
FA-1
FS-1
n
CLn
MA-n
MS-n
FA-n
FS-n
105-1
105
MEMORY HEAP AREA
SYSTEM HEAP AREA
DIVIDED HEAP AREA
DIVIDED HEAP AREA
DIVIDED HEAP AREA
105a0
105a
105n